

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method for serially aligning database transactions comprising at least two databases coupled to their associated database management systems, comprising the steps of:

initiating a first transaction in the first database;
linking into the first transaction at least one transaction trigger that is a deferred database operation defined in the first transaction and specified to be executed after successful completion of the ~~including attributes into said first~~ transaction;

ending said first transaction in the first database;
firing at least one said trigger in at least one first database; and

immediately after the ending and firing steps are completed, the deferred database operation initiating at least one second transaction in the first database, the second transaction initiating to invoke a remote database transaction in at least one second database ~~according to at least some of the attributes in the trigger.~~

2. (currently amended) A method according to claim 1, wherein the ~~said trigger is a deferred database operation defined~~ for at least one data manipulation operation.

3. (previously presented) A method according to claim 1, wherein the execution of the second transaction is blocked until the said trigger fires.

4. (previously presented) A method according to claim 1, wherein a database system comprises at least one master database and at least one replica database, and the data synchronization between the master and replica databases is master-initiated.

5. (previously presented) A method according to claim 1, wherein the transactionally consistent set of data in a database comprises configuration data.

6. (previously presented) A method according to claim 5, wherein the device changes its configuration to reflect the changed data right after the data has committed in the database.

7. (previously presented) A method according to claim 1, wherein the related software processes, like other database server or a client application, are informed about transactional changes by the data management server.

8. (previously presented) A method according to claim 1, wherein the method executes tasks and operations in a database transaction context.

9. (previously presented) A method according to claim 1, wherein any of the said transactions are executed in separate database connections or in a shared connection with another said transaction or another transaction.

10. (previously presented) A method according to claim 1, wherein the method is compatible with at least one of the following communication specifications: TCP/IP, CDMA, GSM, HSCSD, GPRS, WCDMA, EDGE, UMTS, Bluetooth, Teldesic, Iridium, Inmarsat, WLAN, DIGI-TV and imode.

11. (previously presented) A method according to claim 1, wherein the method is compatible with at least one of the following operating systems and is used in at least one terminal including an application, replica database of the database system Unix, MS-Windows, EPOC, NT, MSCE, Linux, PalmOS, GEOS, VxWorks, Pocket PC and any upgrade of these.

12. (previously presented) A method according to claim 1, wherein at least one of the following operating systems is used in at least one server including an application master database of the database system: Unix, MS-Windows, VxWorks, NT and Linux and any upgrade of these.

13. (canceled)

14. (currently amended) A method according to claim [[13]] 1, wherein the set of data of the second transaction

comprises data for performing push-style or push-pull-style synchronization.

15-16. (canceled)

17. (currently amended) A method according to claim [[13]] 14, wherein a database system comprises at least one master database and at least one replica database, the push synchronization data between the master and replica databases is master-initiated and pull synchronization data between the master and replica databases is replica-requested.

18-25. (canceled)

26. (currently amended) An arrangement for serially aligning database transactions comprising:

at least two databases and the associated database management system, a first transaction being initiated in a first one of said two databases,

means for linking into the first transaction at least one transaction trigger that is a deferred database operation defined in the first transaction and specified to be executed after successful completion of ~~including attributes into the~~ first transaction in the first database, and

means for said deferred database operation initiating at least one second transaction in said first database, the second transaction initiating ~~to invoke~~ a remote database transaction in ~~at least one second database according to at least~~

~~some of the attributes in the trigger~~ a second one of said two databases immediately after said first transaction is ended in the first database and said trigger is fired in at least one first database.

27. (previously presented) An arrangement according to claim 26, comprising at least one master database and one replica database coupled to associated database management systems.

28. (previously presented) An arrangement according to claim 26, wherein the transactionally consistent set of data in a database comprises system configuration data.

29. (previously presented) An arrangement according to claim 26, wherein at least the second database can be part of a router coupled to the application.

30. (previously presented) An arrangement according to claim 26, further comprising a storage medium that is one of a memory and a disk.

31. (previously presented) An arrangement according to claim 26, wherein the arrangement and/or database system is compatible with at least one of the following communication specifications: TCP/IP, CDMA, GSM, HSCSD, GPRS, WCDMA, EDGE, UMTS, Bluetooth, Teldesic, Iridium, Inmarsat, WLAN, DIGI-TV and imode.

32. (previously presented) An arrangement according to claim 26, wherein at least one of the following operating systems

and is used in at least one terminal including an application, replica database of the database system Unix, MS-Windows, EPOC, NT, MSCE, Linux, PalmOS, GEOS, VxWorks, Pocket PC and any upgrade of these.

33. (previously presented) An arrangement according to claim 26, wherein at least one of the following operating systems is used in at least one server including an application master database server and/or the schema management node: Unix, MS-Windows, VxWorks, NT and Linux and any upgrade of these.

34. (currently amended) An arrangement according to claim 26, wherein said ~~for serially aligning database transactions comprising~~

~~at least two databases and the associated database management system, comprising means for linking at least one transaction trigger including attributes into the first transaction in the first database, and~~

~~means for initiating at least one second transaction in the first database to synchronize~~ synchronizes ~~data in at least one the second database from at least one~~ with data in said first database according to at least some of the attributes in the trigger immediately after said first transaction is ended in the first database and said trigger is fired in at least one first database.

35. (previously presented) An arrangement according to claim 34, wherein the set of data comprises data of push or push-pull synchronization.

36-42. (canceled)

43. (new) The method of claim 1, further comprising the step of executing the remote database transaction in the second database in response to the second transaction, wherein the remote database transaction is a request to the first database to transfer data to the second database.